

# 19. Модели, моделиране и проектиране (запознаване)

Проф. д-р Емил Хаджиколев

# Модел

- Моделът е опростено представяне на реален или абстрактен обект, система, явление, процес.
  - В частност, една софтуерна система е модел на абстрактна или реална система, на нейните физически и функционални характеристики.
- Моделът възпроизвежда някаква част от действителността, като съдържа съществените и игнорира несъществените детайли;
  - Кое е съществено и кое несъществено зависи от предназначението на модела и конкретната реализация.
- За разлика от реалната система, моделът е винаги достъпен за наблюдаване, анализ и изследване.

# Моделиране

- Моделирането е метод на познанието, състоящ се в създаване и изследване на модели.
- Всеки обект/система има голям брой различни свойства.
- В процеса на построяване на модела се отделят главните, най-съществените за решаването на конкретната задача свойства.
  - Напр. в една софтуерна система обектите се моделират само с малка част от реалните им характеристики. За студент може да са необходими само:
    - физически характеристики като: име, ЕГН, факултетен номер, списък с оценки и др.;
    - функционални характеристики: дата на раждане (от ЕГН-то се взима), среден успех (от списъка с оценки) и др.
      - Защо датата на раждане да се взима от ЕГН-то, а да не се пише като физическа характеристика?! (Не е забранено и понякога се допуска, но...) Излишеството/дублирането на информация (в данните) понякога води до проблеми – ако има грешка в една от стойностите, ще възникне въпрос коя е вярната.

# Проектиране на софтуерни системи

- Важна част от софтуерния процес е проектирането на различни аспекти на системата.
- Проектирането
  - подпомага разработката на системата;
  - подобрява комуникацията между членовете на екипа и с клиентите;
  - служи за документация.
- При проектирането се използват различни диаграми за нагледно представяне на аспектите на системата.

# Видове диаграми

- **Архитектура на системата** – описват слоеве, модули и компоненти на приложението и основните комуникационни връзки и протоколи между елементите.
- **Процеси** – представлящи основните събития, дейности и взаимодействия между компоненти и потребители на системата.
- **Обекти и взаимовръзките им (Entity-Relationship)**, използвани най-често за моделиране на бази данни – описват основните физически характеристики на същностите и взаимовръзките между тях, които могат да бъдат от вид „едно към едно“, „едно към много“, „много към много“ (означавани като 1:1, 1:n, n:m);
- **Алгоритми** – описват потока данните и дейностите, които се изпълняват върху тях.
- и др.

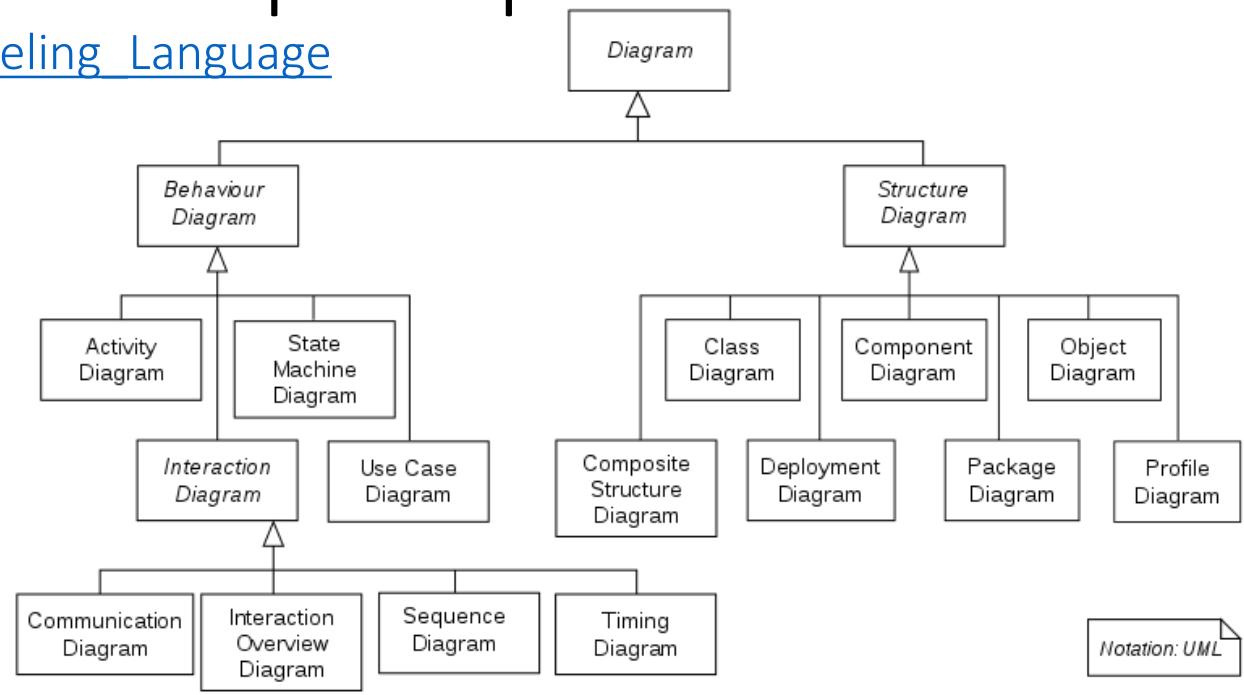
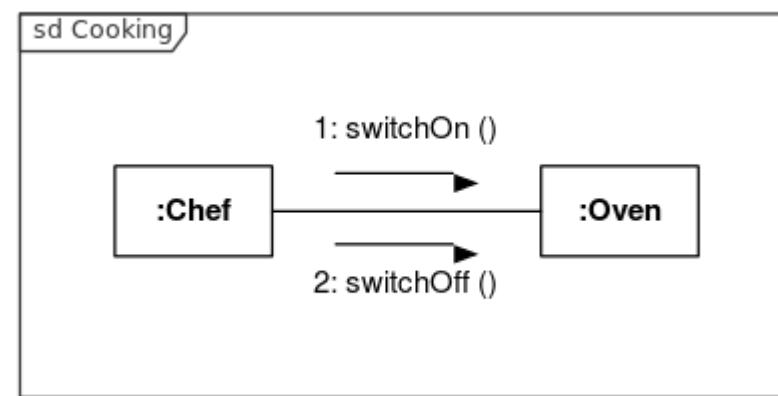
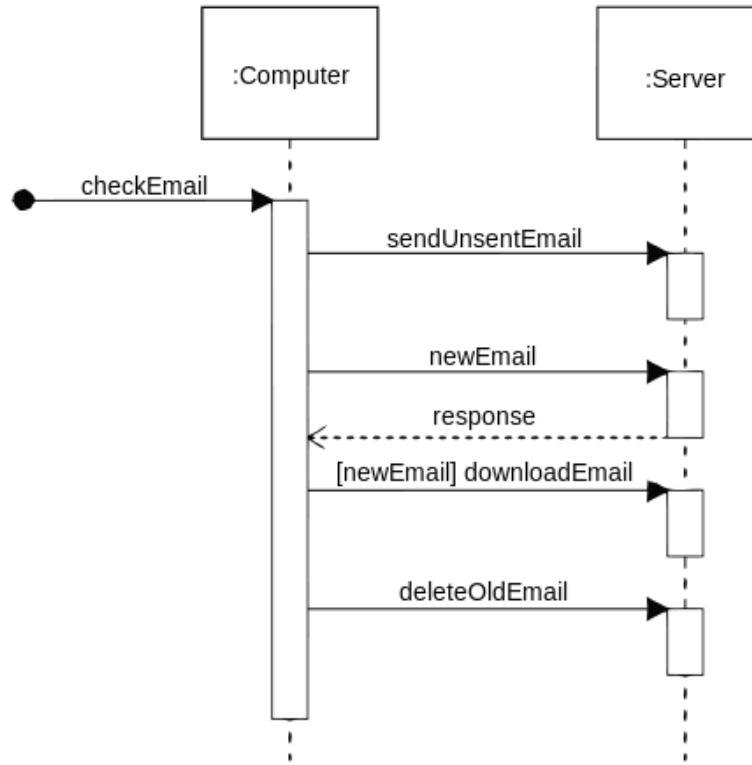
# UML

- **UML (Unified Modeling Language)** – език, с който може да се моделират почти всякакви аспекти в процеса на проектиране на софтуер.
- За моделиране на различни аспекти се предпочитат специализирани инструменти, предлагащи специфични видове графични елементи.
- <https://www.uml.org>.

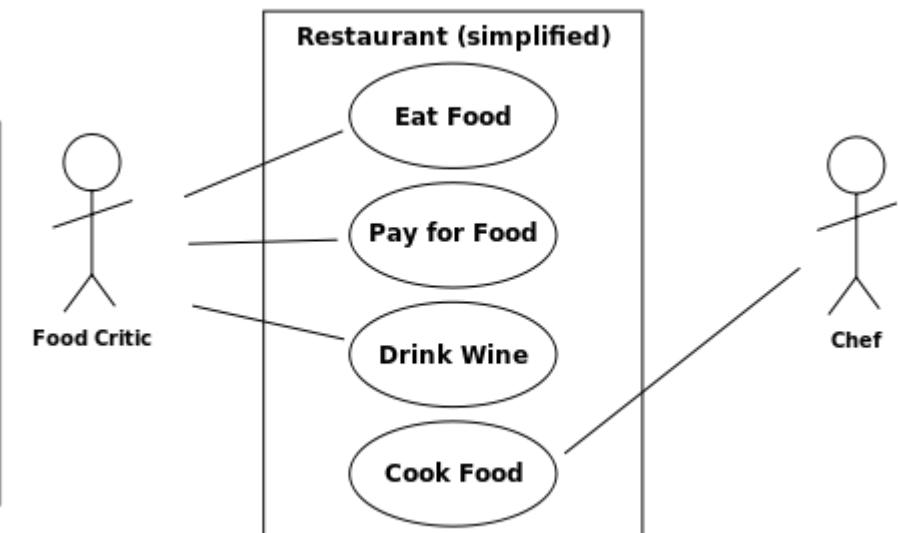
*Следващите примерни диаграми представят част от възможностите на езиците и нотациите. (Не са представени с цел разбиране и разучаване)*

# UML – видове диаграмми и примери

от [https://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language)



Notation: UML



# Езици и нотации за моделиране на архитектури

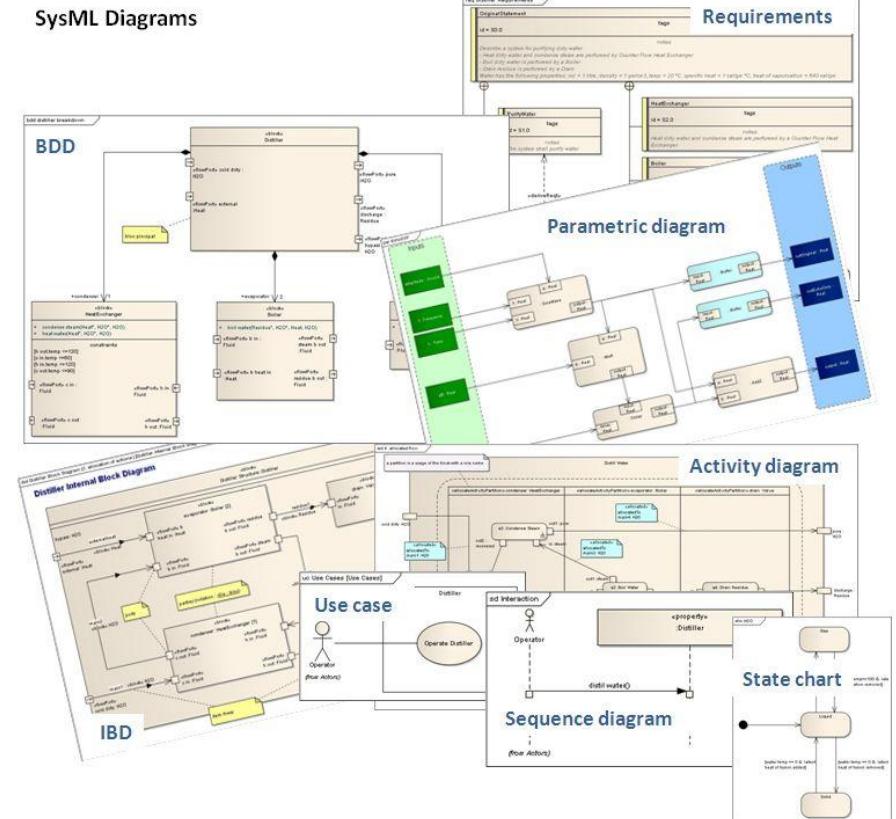
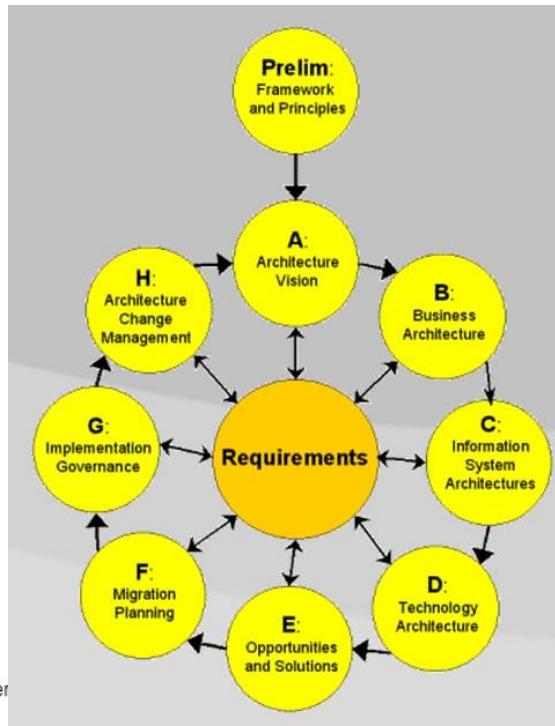
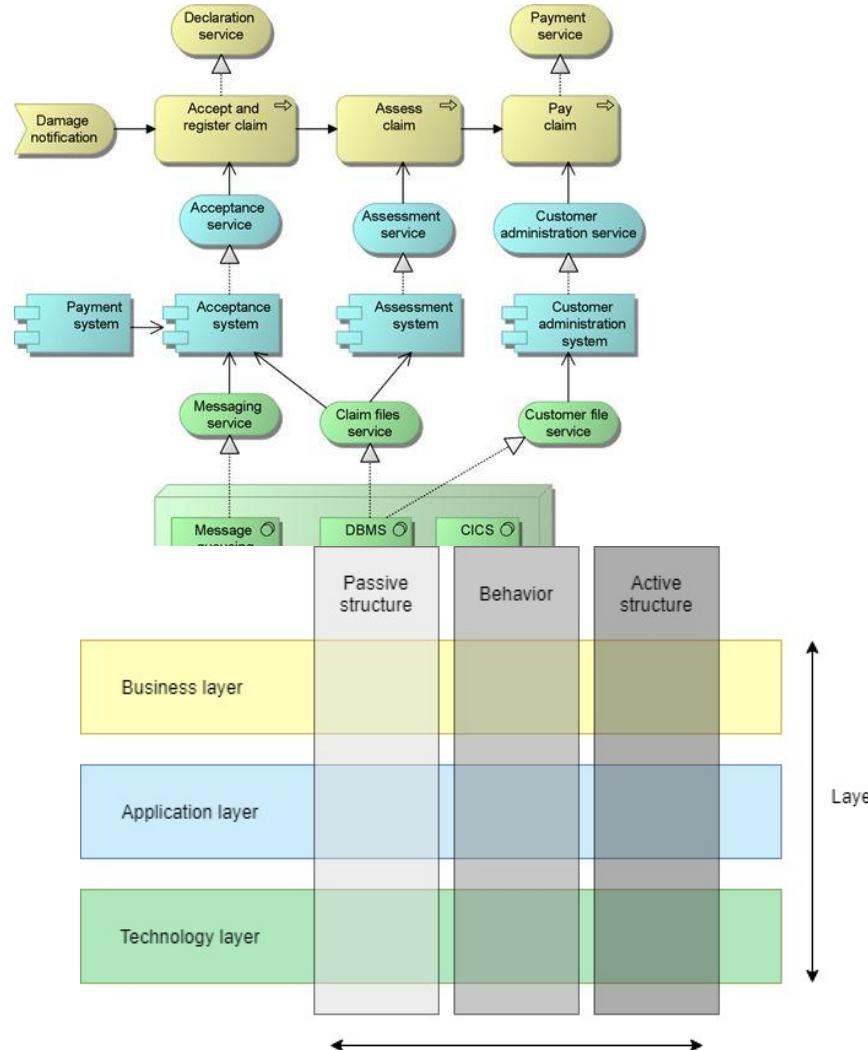
- **UML**: Class Diagrams, Deployment Diagrams, Component Diagrams, Communication Diagrams и др.
- **Archimate** – език за моделиране на бизнес и IT архитектури.
- **SysML (Systems Modeling Language)** – разширение на UML за моделиране на системи.
- **TOGAF (The Open Group Architecture Framework)** – методология/рамка за моделиране на предприятие и технологични архитектури.
- и др.

# Архитектури – методологии, аспекти и примери

<https://en.wikipedia.org/wiki/ArchiMate>

[https://en.wikipedia.org/wiki/The\\_Open\\_Group\\_Architecture\\_Framework](https://en.wikipedia.org/wiki/The_Open_Group_Architecture_Framework)

[https://en.wikipedia.org/wiki/Systems\\_modeling\\_language](https://en.wikipedia.org/wiki/Systems_modeling_language)



# Езици и нотации за моделиране на процеси

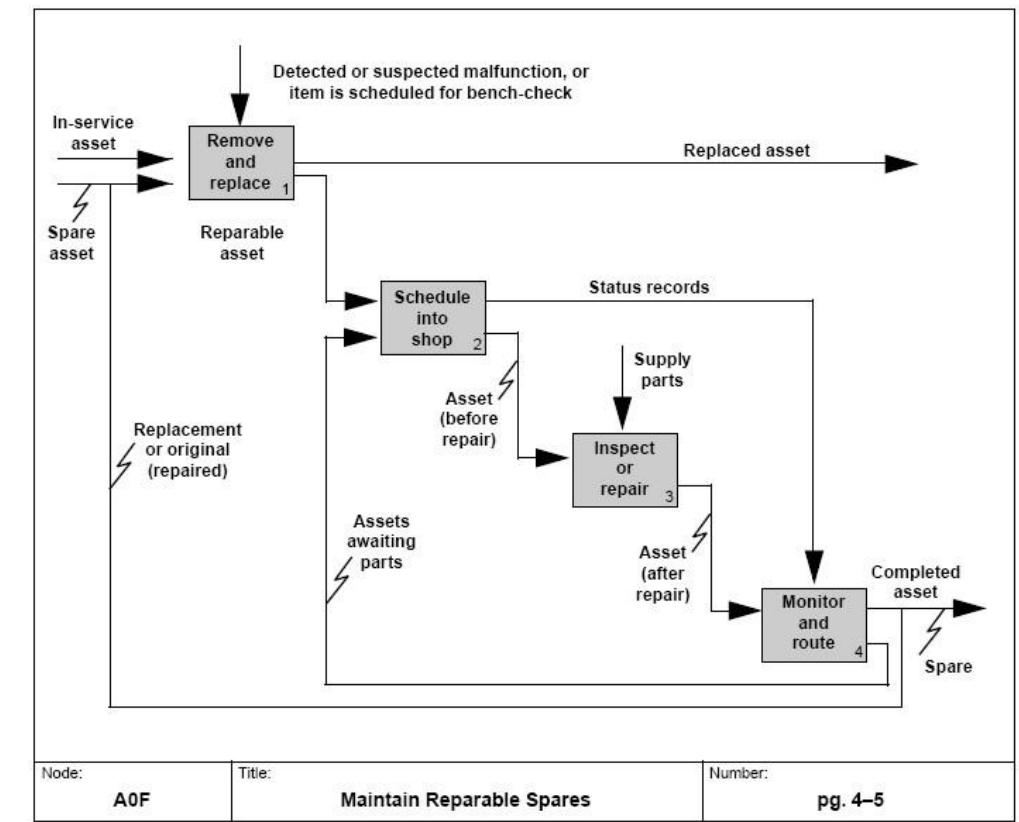
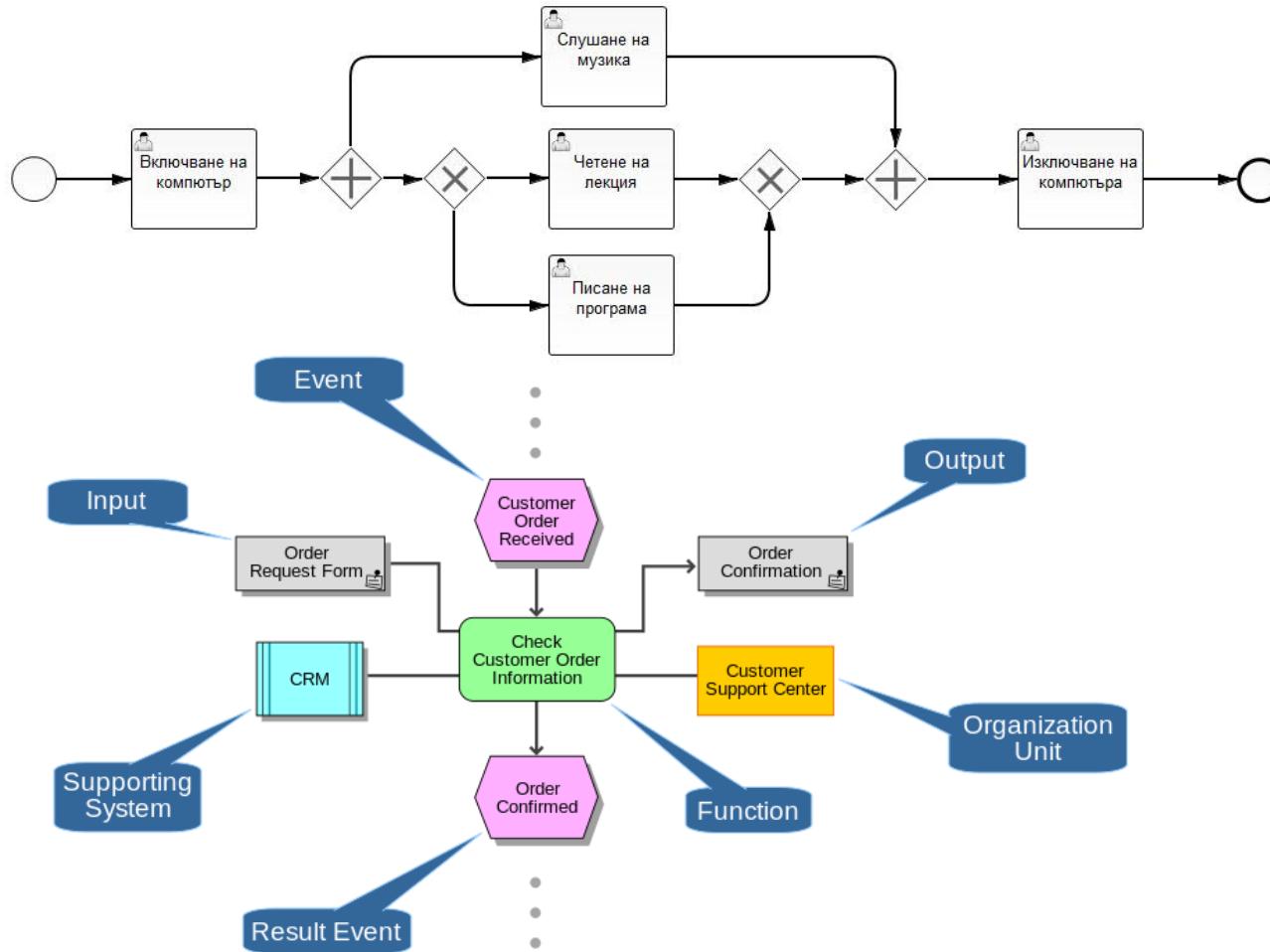
- **UML**: Activity Diagrams, Sequence Diagrams...
- **BPMN** (Business Process Model and Notation) – за моделиране на бизнес процеси.
- **EPC** (Event-driven Process Chain) – нотация за моделиране на процеси, базирана на събития.
- **IDEF0** – нотация за моделиране на функционални процеси.
- и др.

# Процеси – примери

[https://en.wikipedia.org/wiki/Business Process Model and Notation](https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation)

[https://en.wikipedia.org/wiki/Event-driven process chain](https://en.wikipedia.org/wiki/Event-driven_process_chain)

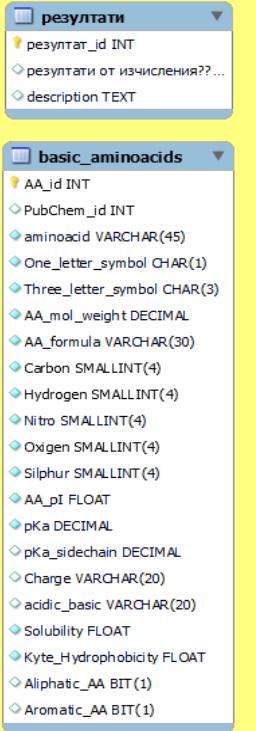
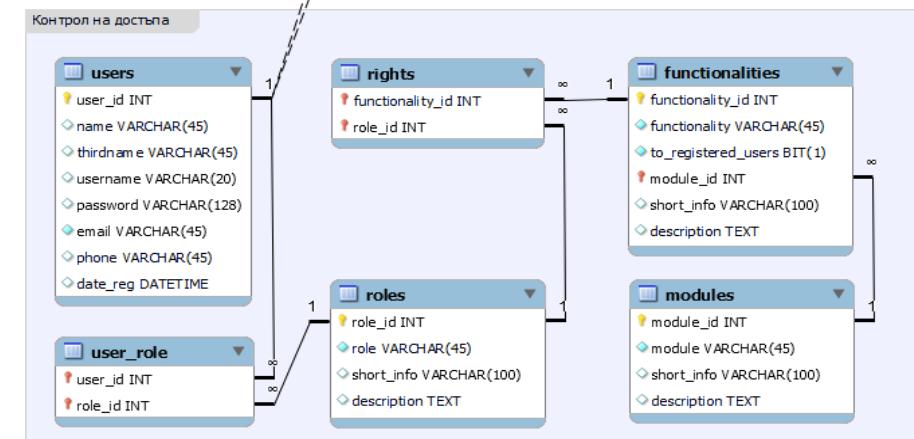
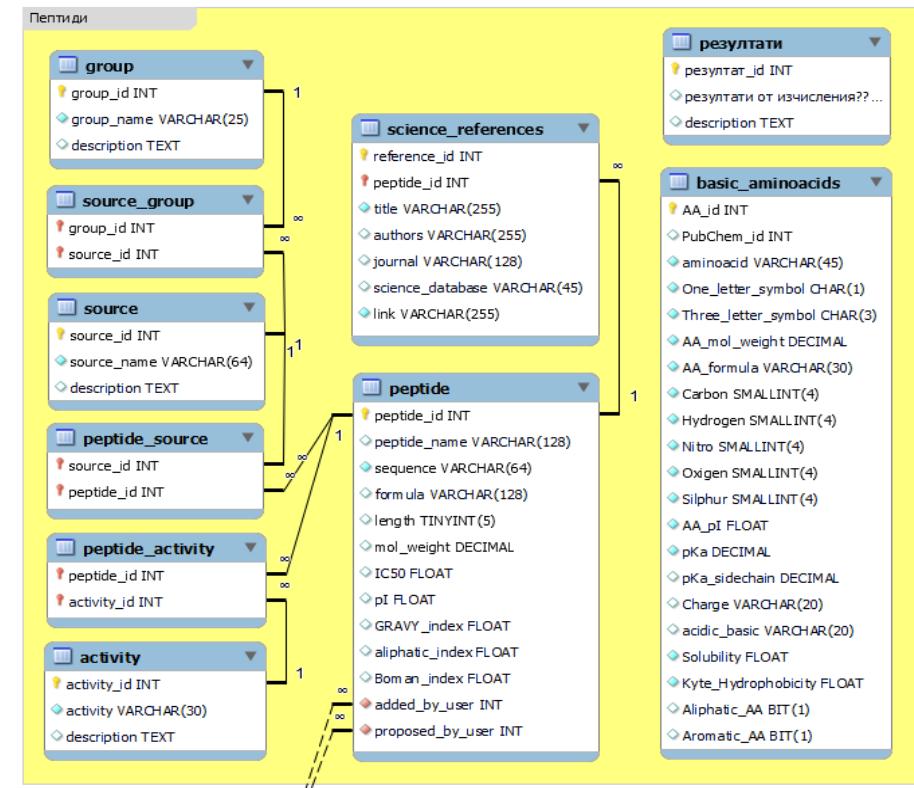
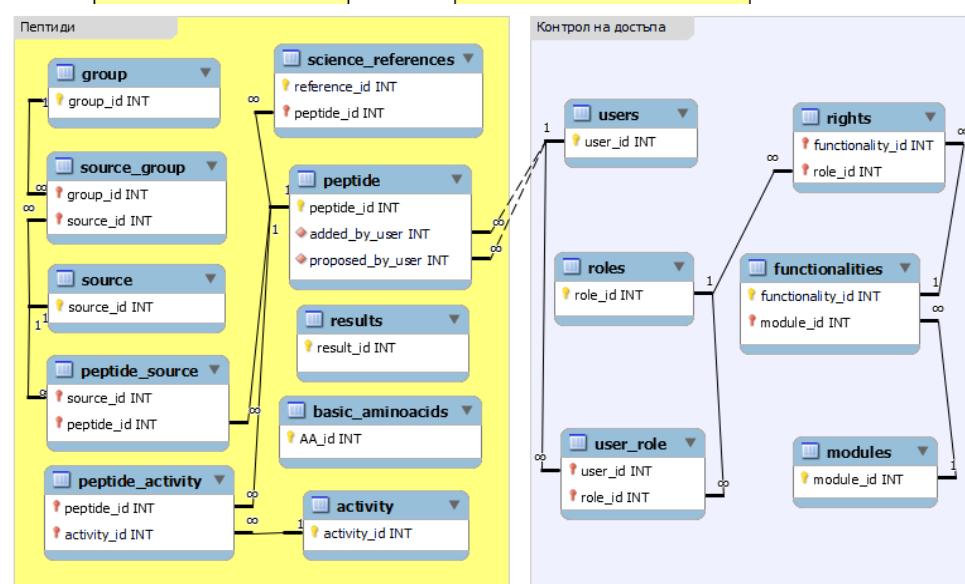
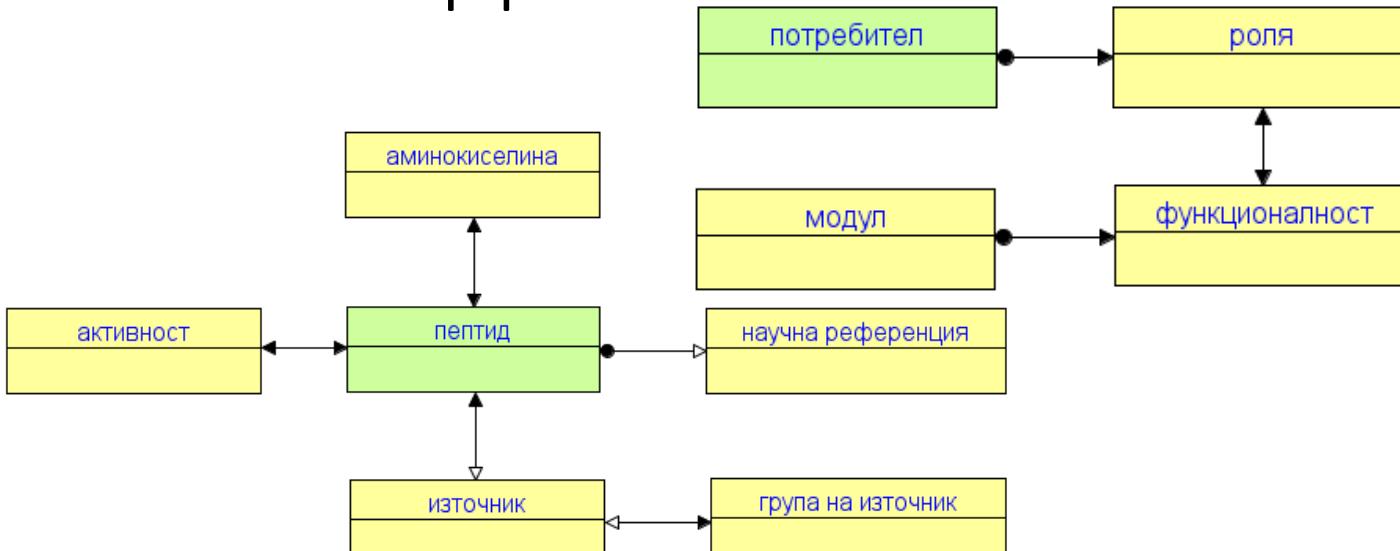
<https://en.wikipedia.org/wiki/IDEFO>



# ER модели

- Нотациите за представяне на ER модели се различават най-вече по различни символи за представяне на същностите и връзките между тях.
- Описват се базите данни на концептуално, логическо и физическо ниво. В някои среди за моделиране моделите могат да бъдат създадени и преобразувани за различни видове бази от данни – релационни (най-често използвани) и различни видове не-релационни.
- Нотации: Chen Notation, Crow's Foot Notation, IDEF1X, Barker's Notation, UML Class Diagrams.

# ER модели

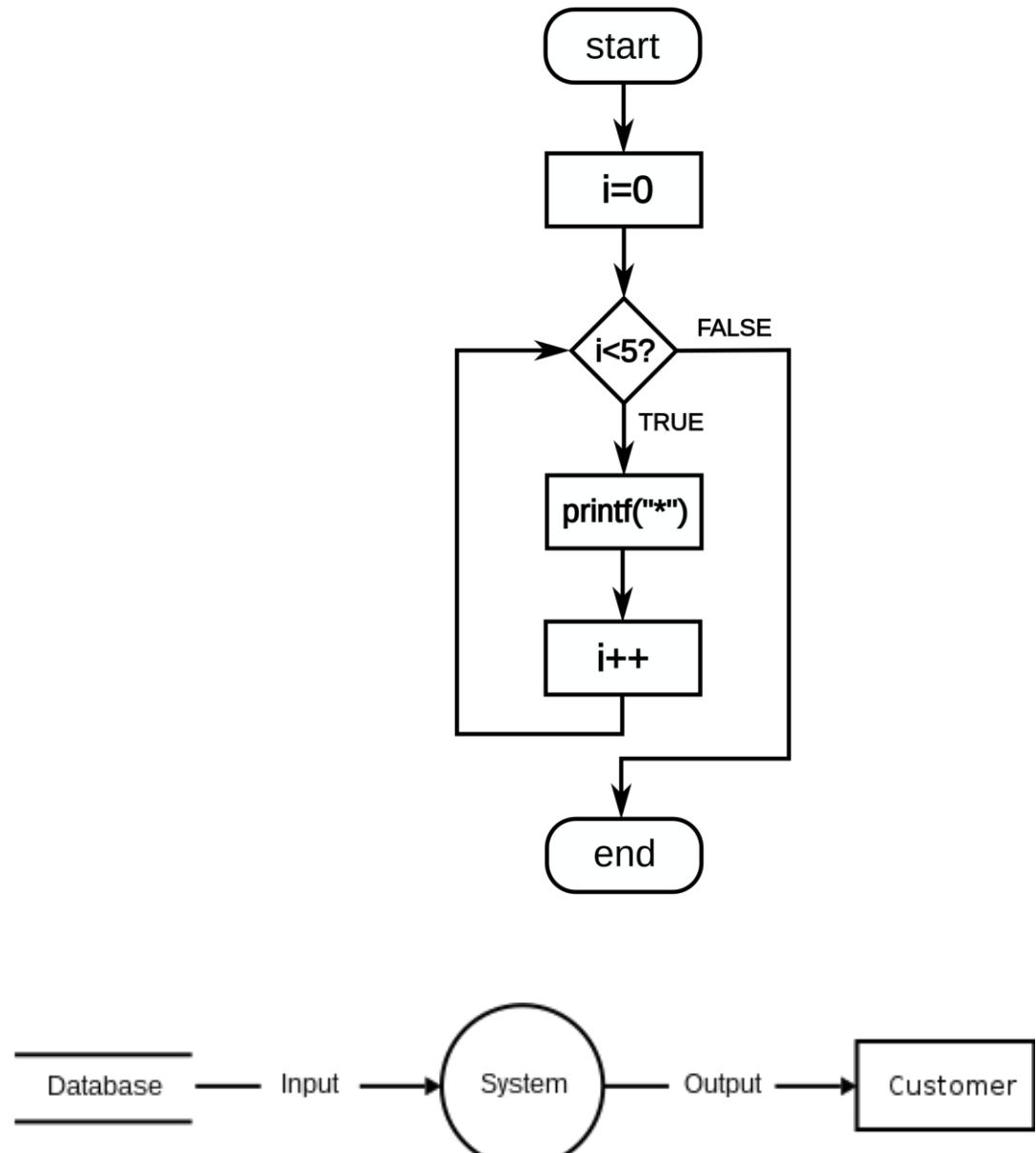
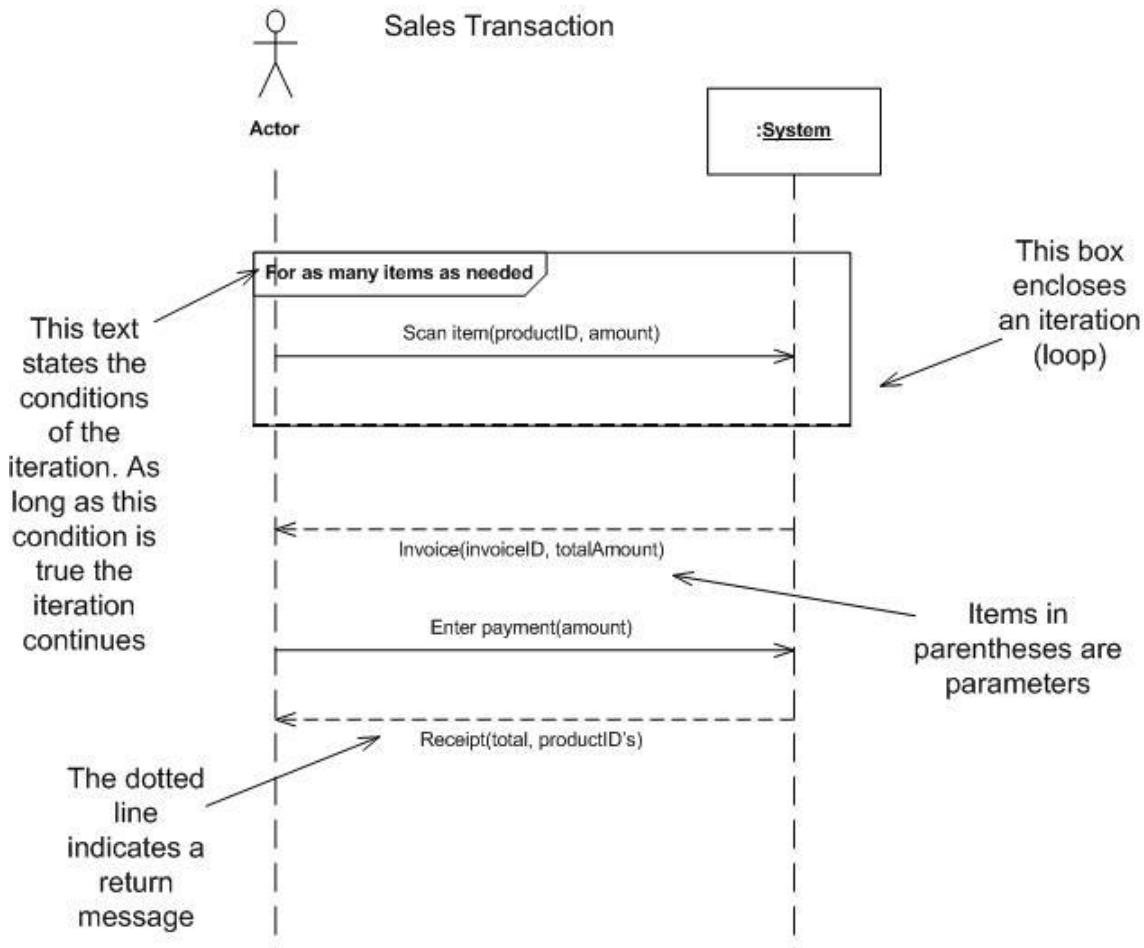


# Диаграми за представяне на алгоритми

- **UML**: State Diagrams, Activity Diagrams, Sequence Diagrams и др.
- **Data Flow Diagrams** – показва потока на данни през процес или система.
- **Flowchart (блок-схеми)** – стандартен подход за описание стъпките на алгоритми, а също и за описание на процеси и работни потоци.
- и др.

# Алгоритми – примери

[https://en.wikipedia.org/wiki/Data-flow\\_diagram](https://en.wikipedia.org/wiki/Data-flow_diagram)  
<https://en.wikipedia.org/wiki/Flowchart>  
[https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram)



# Някои среди за моделиране

- Някои среди са специализирани за конкретни езици и нотации.
  - При повечето от тях се прави контрол за коректно свързване на графичните елементи – подобно на всеки език може да се определят синтактични правила за връзка между графични елементи.
- Среди, които позволяват създаването на разнообразни диаграми обикновено няма синтактичен контрол и подсказки.
  - В този случай проектантът трябва да знае много добре какво иска да направи, да познава елементите и правилата за свързването им.
- Популярни среди
  - бесплатни: StarUML, PlantUML, Bizagi Modeler, <https://bpmn.io/>, <https://draw.io/>, Oracle Data Modeler, MySQL Workbench
  - платени: Microsoft Visio Drawing...
  - и мн. др. платени, бесплатни или с trial лицензи; онлайн или десктоп приложения.